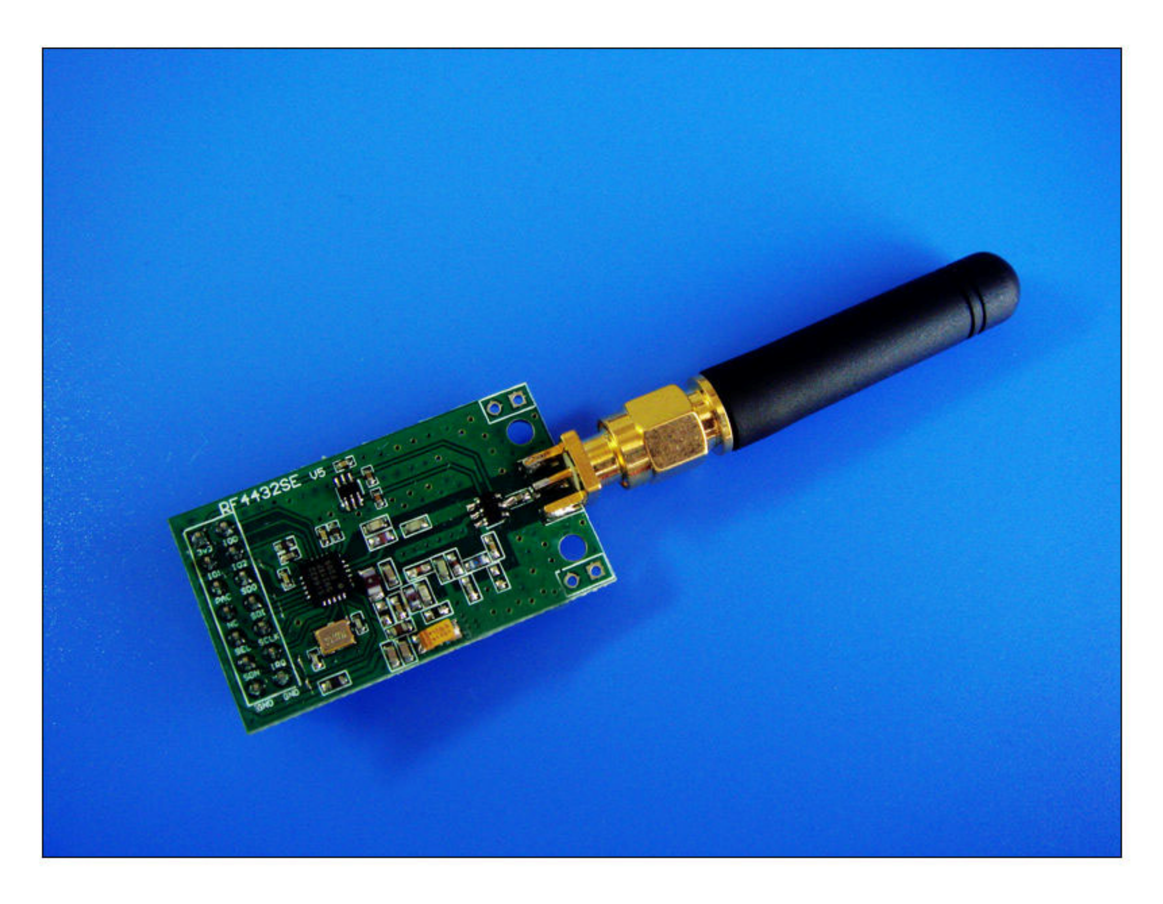
Si4432 无线模块

用户手册



RF4432SE 模块尺寸(38mm X 22mm，不包括 SMA 头尺寸)

联系电话：13704018223 陈 工 E-mail: chj_006@sina.com
在线咨询：QQ:35625400 474882985 MSN:1188mm88@hotmail.com

目　录

产品简介

本模块采用了 Silicon Labs 的 Si4432 作为无线收发芯片，是一块完整的、体积小巧的、低功耗无线收发模块。SI4432 由 Silicon 公司 09 年推出的 Silicon Labs EZRadioPRO 系列 ISM 频段无线芯片之一,可工作在 240-960 MHZ 频段范围内，且最大输出功率可以达到+20DBm，这一定程度上满足了射频信号放大的开发难的问题。再最大功率设置条件下，开阔通信距离可达上千米。模块集成了所有射频相关功能和器件，用户不需要了解无线电的相关知识，就可以使用本模块轻易的开发出性能稳定可靠的无线相关产品。

基本特点

频率范围：　240—960MHZ

接收灵敏度：-115 dBm（波特率 9.6Kbps）

通讯波特率：1—128kbps

发射功率：　11—20dBm

供电电压：　1.8—3.6v

在关断模式下电流为：10nA

在休眠模式下电流为：　300nA

FSK 频偏可编程(15~240KHz)

接收带宽可编程(67~400KHz)

具有自动斜率控制功率放大器和频谱整形

嵌入式天线多样,空旷通讯距离 800 米以上(波特率 9.6Kbps)

SPI 兼容的控制接口，低功耗任务周期模式，自带唤醒定时器

片内集成电池低电量检测、温度传感器、电压调节器

低的接收电流(18.5mA)，最大发射功率的电流(73mA)

典型主要应用

车辆监控、遥控、遥测、水文气象监控

无线标签、身份识别、非接触 RF 智能卡

小型无线网络、无线抄表、门禁系统、小区传呼

工业数据采集系统、无线 232 数据通信、无线 485/422 数据通信、

小型无线数据终端、安全防火系统、无线遥控系统、生物信号采集、

模块接口说明

RF4432

定义	引脚	引脚	定义
VCC3.3	1	2	IO0
IO1	3	4	IO2
PAC	5	6	SDO
NC	7	8	SDI
SEL	9	10	SCLK
SDN	11	12	IRQ
GND	13	14	GND

引脚功能说明

引脚	定义	类型	说明
1	3V3	3.3V 电压	电源电压
2	IO0	通用 IO	可不用
3	IO1	通用 IO	可不用
4	IO2	通用 IO	可不用
5	PAC	收发切换	收发方式切换
6	SDO	数据输出	串行数据输出
7	NC	无	悬空
8	SDI	数据输入	串行数据输入

9	SEL	片选	串行片选
10	SCLK	时钟	串行时钟输入
11	SDN	关闭引脚	SDN=1，关闭芯片工作
12	IRQ	中断	中断信号的产生
13	GND	电源地	和系统共地
14	GND	电源地	和系统共地

备注

1.VCC 引脚的电压范围为1.9-3.6V 之间，不能在这个区间之外，如超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右；

2.硬件没有集成SPI功能的单片机也可以控制本模块，用普通单片IO口模拟 SPI 时序进行读写操作即可；

3. 模块接口采用标准2.54mmDIP插针，13 脚、14 脚为接地脚,需要和系统电路的逻辑地连接起来；

4.与 51 系列单片机 P0 口连接时候，需要加 10K 的上拉电阻,与其余口连接不需要。其他系列的5V单片机，如AVR、PIC，请参考该系列单片机 IO 口输出电流大小，如果超过 10mA，需要串联2-5K电阻分压，否则容易烧毁模块！如果是 3.3V 的MCU，可以直接和IO口连接。

模块电气参数

参数	数值	单位
工作电压	1.8-3.6	V
最大发射功率	20	dBm
最大数据传输率	128	kbps
温度范围	-40 to +85	
典型灵敏度	-107（40kbps条件）	dBm

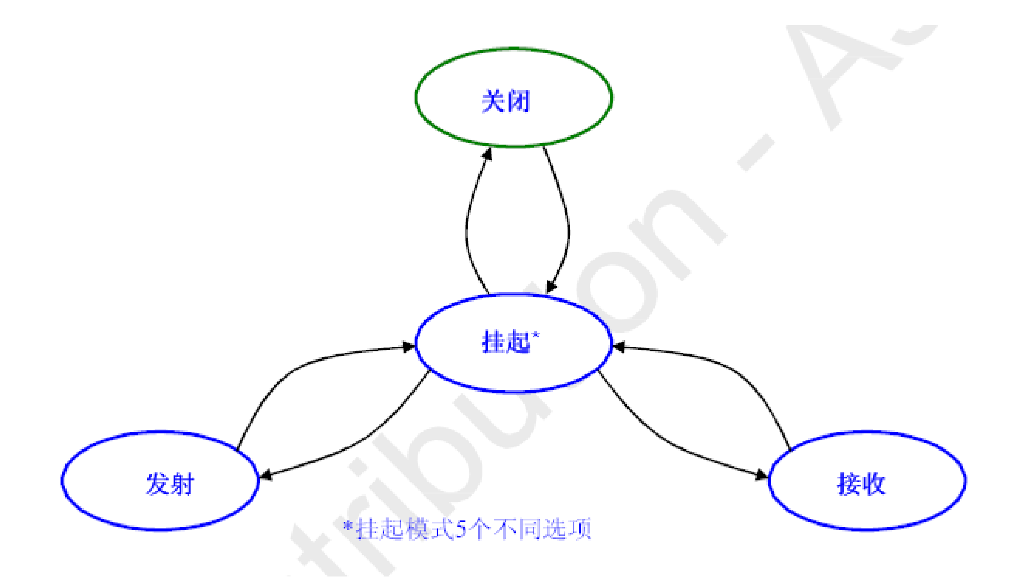
工作方式

工作模式

Si4432 主要有关闭模式、挂机模式、发射模式和接收模式组成，关闭状态下可以降低功耗。各模式之间切换必须先进入挂起状态然后再切换，以下是各模式工作参数；

状态 / 模式	xtal	pll	wt	LBD or TS	响应时间: TX	响应时间: RX	状态/模式电流 [uA]
关闭状态	X		X X	X	4.21msec	4.21msec	10nA
挂起状态							
待机模式	0	0	0	0	1.21msec	1.21msec	300nA
睡眠模式	0	0	1	0			600nA
传感器模式	0	0	X	1			245uA
预备模式	1	0	X	X	210usec	210usec	600uA
调谐模式	1	1	X	X	200usec	200usec	8mA
发射状态	1		1 X	X	NA	200usec	60mA Æ +20dBm
接收状态	1		1 X	X	200usec	NA	17mA

以下为工作模式切换示意图



Si4432 数据传输方式

Si4432 数据传输传输有 FIFO Mode、Direct Mode 和 PN9 Mode 三种传输模式

FIFO Mode

在 FIFO Mode 下，使用片内的先入先出堆栈区来发送和接收数据。对 FIFO 的操作是通过 SPI 对 07H 寄存器的连续读或者写进行的。在 FIFO Mode 下，Si4432 自动退出发送或者接收状态，当相关的中断信号产生，并且自动处理字头和 CRC 校验码。在接收数据时，自动把字头和 CRC 校验码移去。在发送数据时，自动加上字头和 CRC 校验码。

Direct Mode（不建议使用）

在直接收发模式下，Si4432 如传统的射频收发器一样工作。

PN9 Mode（不建议使用）

在这种模式下的 Tx 数据是内部产生使用伪随机（ PN9 序列）位发生器。这种模式的目的是用作测试模式不断观察调制频谱，而不必负载/提供数据。

模块参数配置

配置 Si4432 模块是通过 SPI 方式进行的，可配置为 FIFO 方式和直接方式，我们推荐 Si4432 工作于 FIFO 收发模式，这种工作模式下，系统的程序编制会更加简单，并且稳定性也会更高，因此，下文着重介绍把 Si4432 配置为 FIFO 收发模式的器件配置方法。其中配置 Si4432 主要包括载波频率、调制方式、数据发送速率、CRC 校验、前导码、同步字、数据头、地址等，具体可参考 Si4432(IA4432) Register Settings_RevV-v16 寄存器配置工具进行。

程序设计参考

使用Si4432模块无需掌握任何专业无线或高频方面的理论，读者只需要具备一定的C语言程序基础即可。本文档没有涉及到的问题，读者可以参考Si4432官方手册或向我们寻求技术支持。同时，为便于用户开发，我们提供配套评估套件，为产品开发保驾护航，使无线应用开发大大加速，并避免不必要的误区。

SPI 读操作函数

```
unsigned char SPI_Read(void)
{
    unsigned char i,rxdata;
    rxdata = 0x00;
    for (i = 0;i < 8;i++)
    {
        rxdata = rxdata<<1;
        RF4432_SCLK=1;
        RF4432_SDO=1;
        if (RF4432_SDO==1)
             //读取最高位，保存至最末尾，通过左移位完成整个字节
        {
            rxdata |= 0x01;
        }
        else
        {
            rxdata &= ~0x01;
        }
        delay_10us(2);
        RF4432_SCLK=0;
        delay_10us(2);
     }
     return rxdata;
}
```

SPI 写操作函数

```
void SPI_Write(unsigned char txdata)
{
    unsigned char i;
    for (i = 0;i < 8;i++)
    {
        if (txdata&0x80)  //总是发送最高位
        {
        RF4432_SDI=1;
        }
        else
        {
        RF4432_SDI=0;
        }
```

```
        RF4432_SCLK=1;
        txdata = txdata<<1;
        RF4432_SCLK=0;
    }
}
```

RF4432 寄存器读操作函数

```
unsigned char  RF4432_ReadReg(unsigned char  addr)
{
    unsigned char value;
    RF4432_SEL=0;
    SPI_Write(addr|RR);
    value = SPI_Read();
    RF4432_SEL=1;
    return value;
}
```

RF4432 寄存器写操作函数

```
void  RF4432_WriteReg(unsigned char  addr, unsigned char value)
{
    RF4432_SEL=0;
    SPI_Write(addr|WR);
    SPI_Write(value);
    RF4432_SEL=1;
}
```

RF4432 射频芯片初始化函数

```
void RF4432_Init(void)
{
    RF4432_SDN=0;     //允许 RF4432 工作
    delay_ms(20);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_1, 0x80);
    //软件复位，详细请见 P101 页
    delay_ms(10);
    RF4432_ReadReg(INTERRUPT_STATUS_1);
    RF4432_ReadReg(INTERRUPT_STATUS_2);
    RF4432_WriteReg(INTERRUPT_ENABLE_1, 0x00);
    RF4432_WriteReg(INTERRUPT_ENABLE_2, 0x00);
```

```
    RF4432_WriteReg(IF_FILTER_BANDWIDTH, 0x8b);
    RF4432_WriteReg( CLOCK_RECOVERY_OVERSAMPLING_RATIO, 0x2c);
    RF4432_WriteReg( CLOCK_RECOVERY_OFFSET_2, 0x20);
    RF4432_WriteReg( CLOCK_RECOVERY_OFFSET_1, 0x6d);
    RF4432_WriteReg( CLOCK_RECOVERY_OFFSET_0, 0x3a);
    RF4432_WriteReg( CLOCK_RECOVERY_TIMING_LOOP_GAIN_1, 0x00);
    RF4432_WriteReg( CLOCK_RECOVERY_TIMING_LOOP_GAIN_0, 0x18);
    RF4432_WriteReg(RSSI_THRESHOLD_FOR_CLEAR_CHANNEL_INDICATOR, 0xf0);
    RF4432_WriteReg( DATA_ACCESS_CONTROL, 0x88);
    RF4432_WriteReg( HEADER_CONTROL_1, 0x8c);
    RF4432_WriteReg( HEADER_CONTROL_2, 0x0a);
    RF4432_WriteReg( PREAMBLE_LENGTH, 0x08);
    RF4432_WriteReg(  TRANSMIT_PACKET_LENGTH,   RF4432_TxRxBuf_Len  );
    //发送数据包长度设置
    RF4432_WriteReg( RECEIVED_PACKET_LENGTH, RF4432_TxRxBuf_Len );
    //接收数据包长度设置
    RF4432_WriteReg( HEADER_ENABLE_3, 0x00);
    RF4432_WriteReg (HEADER_ENABLE_2, 0x00);
    RF4432_WriteReg( HEADER_ENABLE_1, 0x00);
    RF4432_WriteReg( HEADER_ENABLE_0 ,0x00);
    RF4432_WriteReg(AGC_OVERRIDE_2, 0x0b);
    RF4432_WriteReg(TX_POWER,0x03);
    RF4432_WriteReg(TX_DATA_RATE_1, 0x0A);    //通信速率 40kbps
    RF4432_WriteReg(TX_DATA_RATE_0, 0x3D);
    RF4432_WriteReg(MODULATION_MODE_CONTROL_1, 0x02);
    RF4432_WriteReg(MODULATION_MODE_CONTROL_2, 0x26);
    //设置为 FIFO 模式  FSK  reg71H
    RF4432_WriteReg(FREQUENCY_DEVIATION, 0x40);  //
    RF4432_WriteReg(FREQUENCY_BAND_SELECT, 0x53);//设置频率  reg75H
    RF4432_WriteReg(NOMINAL_CARRIER_FREQUENCY_1, 0x4b);
    //通信频率 433MHZ     reg76H
    RF4432_WriteReg(NOMINAL_CARRIER_FREQUENCY_0, 0x00);//reg77H
    RF4432_WriteReg(TX_FIFO_CONTROL_1,0x3F);
    RF4432_WriteReg(TX_FIFO_CONTROL_2,0x00);
    RF4432_WriteReg(RX_FIFO_CONTROL,0x3F);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERA
TING_FUNCTION_CONTROL_2)|0x01);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERA
TING_FUNCTION_CONTROL_2)&0xFE);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERA
TING_FUNCTION_CONTROL_2)|0x02);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERA
TING_FUNCTION_CONTROL_2)&0xFD);
```

```
}
```

RF4432 设置接收模式函数

```
void RF4432_SetRxMode(void)
{
    RF4432_PAC=0;
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERATING_FUNCTION_CONTROL_2)|0x02);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERATING_FUNCTION_CONTROL_2)&0xFD);
    RF4432_ReadReg(INTERRUPT_STATUS_1);
    RF4432_ReadReg(INTERRUPT_STATUS_2);
    RF4432_WriteReg(INTERRUPT_ENABLE_1, 0x02);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_1,0x05);//接收

}
```

RF4432 数据包接收函数

XTAL
建立时间
PLL T0
PLL CAL
PLLTS
接收包
~1ms
可配置范围：0-70uS，默认=10uS
45us,可以跳过
可配置范围：0-310us，建议 200us


```
unsigned char RF4432_RxPacket(void)
{
    RF4432_IRQ=1;
    if(!RF4432_IRQ)
    {
        if(RF4432_ReadReg(INTERRUPT_STATUS_1)&0x02)
        {
```

```
            return 1;
        }
    }
        return 0;
}
```

RF4432 数据包发送函数

XTAL 建立时间 | PLL TO | PLL CAL | PLL TS | PRE PA RAMP | PA RAMP UP | TXMOD DELAY | 发射包

~1ms

可配置范围0-70us，默认=10us

45us可以跳过

可配置范围：0-310us，建议 200us

6us固定

可配置范围：5-20us，建议 5us

1.5Bits@DR

```
void RF4432_TxPacket(void)
{
    RF4432_PAC=1;
RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERATING_FUNCTION_CONTROL_2)|0x01);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_2,RF4432_ReadReg(OPERATING_FUNCTION_CONTROL_2)&0xFE);
RF4432_WriteBurestReg(FIFO_ACCESS,RF4432_TxRxBuf,RF4432_TxRxBuf_Len);
    RF4432_ReadReg(INTERRUPT_STATUS_1);
    RF4432_ReadReg(INTERRUPT_STATUS_2);
    RF4432_WriteReg(INTERRUPT_ENABLE_1, 0x04);
    RF4432_WriteReg(OPERATING_FUNCTION_CONTROL_1,0x09);//发射
    while(1)
    {
        RF4432_IRQ=1;
        if(!RF4432_IRQ)
        {
         if(RF4432_ReadReg(INTERRUPT_STATUS_1)&0x04) //详细请看 P95 页
```

```
            {
                break;
            }
        }
    }
}
```

无线应用注意事项

(1) 无线模块的 VCC 电压范围为 1.8V-3.6V 之间，不能在这个区间之外，超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右。

(2) 除电源 VCC 和接地端，其余脚都可以直接和普通的 51 单片机 IO 口直接相连，无需电平转换。当然对 3V 左右的单片机更加适用了。

(3) 硬件上面没有 SPI 的单片机也可以控制本模块，用普通单片机 IO 口模拟 SPI 不需要单片机真正的串口介入，只需要普通的单片机 IO 口就可以了，当然用串口也可以了。模块按照接口提示和母板的逻辑地连接起来

(4) 标准 DIP 插针，如需要其他封装接口，或其他形式的接口，可联系我们定做。

(5) 任何单片机都可实现对无线模块的数据收发控制，并可根据我们提供的程序，然后结合自己擅长的单片机型号进行移植;

(6) 频道的间隔的说明：实际要想 2 个模块同时发射不相互干扰，两者频道间隔应该至少相差 1MHZ，这在组网时必须注意，否则同频比干扰。

(7) 实际用户可能会应用其他自己熟悉的单片机做为主控芯片，所以，建议大家在移植时注意以下 4 点：

A:确保 IO 是输入输出方式，且必须设置成数字 IO;

B:注意与使用的 IO 相关的寄存器设置，尤其是带外部中断、带 AD 功能的 IO，相关寄存器一定要设置好;

C:调试时先写配置字，然后控制数据收发

D:注意工作模式切换时间

我们的承诺

最后，欢迎您使用我们的产品，在应用中有技术问题请及时向我们联系，我们会予以技术知道，同时运输中出现产品问题我们会全面责任并予以更换。

愿与您一起走向成功

联系电话：13704018223 陈 工 E-mail: chj_006@sina.com
在线咨询：QQ:35625400 474882985 MSN:1188mm88@hotmail.com